Physics-Informed Neural Networks for Contaminant Transport in Aquifers

Ali Haghighi

Supervised by:

Afshin Ashrafzadeh, François Lehmann, Marwan Fahs

December 18, 2025

Université de Strasbourg

Abstract

This work presents a Physics-Informed Neural Network (PINN) implementation for solving the one-dimensional advection–dispersion equation governing contaminant transport in groundwater aquifers. The complete implementation is available as a Python script: https://github.com/Alioax/pinn/blob/main/pinn_baseline/pinn_baseline.py. The network trains to satisfy the governing partial differential equation and boundary conditions without requiring labeled solution data. The network operates in dimensionless form to improve numerical stability, uses automatic differentiation to compute derivatives, and enforces physics through a weighted loss function combining PDE residual, initial condition, and boundary condition terms. Results demonstrate concentration profiles that capture the advective front propagation and dispersive spreading, compared with the analytical Ogata–Banks solution [1] as a visual reference for post-training comparison. The baseline script and additional implementations are available in the repository: https://github.com/Alioax/pinn.

1 Problem Statement

The governing equation for one-dimensional contaminant transport in a porous medium is the advection–dispersion equation:

$$\frac{\partial C}{\partial t} + U\frac{\partial C}{\partial x} = D\frac{\partial^2 C}{\partial x^2}$$

where $C(x,t)$ is the concentration (kg/m^3). The physical parameters are:

$U = 0.1$ m/day	(advection velocity)
$D = 1.0 \times 10^{-7} \times 86400$ m^2/day	(dispersion coefficient)
$C_0 = 5.0$ kg/m^3	(inlet concentration)

The spatial and temporal domains are:

$$x \in [0, 100] \text{ m}$$
$$t \in [0, 1000] \text{ days}$$

The initial condition is:

$$C(x,0) = 0 \qquad \text{for } x > 0$$

Boundary conditions are:

$$C(0,t) = C_0 \qquad \text{(inlet, Dirichlet)}$$
$$\frac{\partial C}{\partial x}(L,t) = 0 \qquad \text{(outlet, Neumann)}$$

where $L = 100$ m.

2 PINN Implementation

2.1 Parameter Definitions and Dimensionless Formulation

The implementation converts all variables to dimensionless form to improve training stability. The characteristic scales are:

$L = 100$ m	(length scale)
$T = L/U = 1000$ days	(time scale, advective)
$C_0 = 5.0$ kg/m^3	(concentration scale)

Dimensionless variables are defined as:

$$x^* = x/L$$
$$t^* = t/T$$
$$C^* = C/C_0$$

The Péclet number characterizes the relative importance of advection to dispersion:

$$\text{Pe} = \frac{U \times L}{D} \approx 1157.41$$

Substituting the dimensionless variables into the governing equation yields the dimensionless advection–dispersion equation:

$$\frac{\partial C^*}{\partial t^*} + \frac{\partial C^*}{\partial x^*} = \frac{1}{\text{Pe}}\frac{\partial^2 C^*}{\partial x^{*2}}$$

2.2 Neural Network Architecture

The PINN is a fully connected feedforward network with inputs (x^*, t^*) and output C^*. The default architecture consists of 3 hidden layers with 10 neurons per layer, using Tanh activation functions. The network structure is: Input(2) $\to$ Hidden(10) $\to$ Hidden(10) $\to$ Hidden(10) $\to$ Output(1).

2.3 Collocation Point Sampling

The implementation uses 2000 randomly sampled collocation points throughout the domain at each training epoch. For initial conditions, 100 random points are sampled at $t^* = 0$. Boundary conditions are enforced at 100 random points each: inlet at $x^* = 0$ and outlet at $x^* = 1$.

2.4 Automatic Differentiation

Spatial and temporal derivatives are computed using PyTorch's automatic differentiation. The PDE residual is assembled as:

$$R_{\text{PDE}} = \frac{\partial C^*}{\partial t^*} + \frac{\partial C^*}{\partial x^*} - \frac{1}{\text{Pe}}\frac{\partial^2 C^*}{\partial x^{*2}}$$

First derivatives $\partial C^*/\partial t^*$ and $\partial C^*/\partial x^*$ are computed via `grad()` with `create_graph=True` and `retain_graph=True`. The second derivative $\partial^2 C^*/\partial x^{*2}$ is obtained by differentiating $\partial C^*/\partial x^*$ with respect to x^*. The PDE residual R_{PDE} is evaluated at each collocation point and will be used in the loss function formulation described in the next subsection.

2.5 Loss Function Formulation

The total loss combines four components with weights:

$$\mathcal{L}_{\text{total}} = w_{\text{PDE}}\mathcal{L}_{\text{PDE}} + w_{\text{IC}}\mathcal{L}_{\text{IC}} + w_{\text{inlet}}\mathcal{L}_{\text{inlet}} + w_{\text{outlet}}\mathcal{L}_{\text{outlet}}$$

For this baseline implementation, all weights are set to 1, giving equal importance to each loss component. The individual loss components are computed as mean squared errors over their respective sampling points:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{coll}}} \sum_{i=1}^{N_{\text{coll}}} R^2_{\text{PDE}}(x_i^*, t_i^*)$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_{\text{IC}}} \sum_{j=1}^{N_{\text{IC}}} \left[C^*(x_j^*, 0)\right]^2$$

$$\mathcal{L}_{\text{inlet}} = \frac{1}{N_{\text{BC}}} \sum_{k=1}^{N_{\text{BC}}} \left[C^*(0, t_k^*) - 1\right]^2$$

$$\mathcal{L}_{\text{outlet}} = \frac{1}{N_{\text{BC}}} \sum_{k=1}^{N_{\text{BC}}} \left[\frac{\partial C^*}{\partial x^*}(1, t_k^*)\right]^2$$

where $N_{\text{coll}} = 2000$, $N_{\text{IC}} = 100$, and $N_{\text{BC}} = 100$ are the number of sampling points for PDE, initial condition, and boundary conditions, respectively.

2.6 Training Loop

Training uses the Adam optimizer with learning rate 0.001 for 2000 epochs. Each epoch performs: (1) forward pass to evaluate the network at collocation points, (2) PDE residual evaluation via automatic differentiation, (3) loss computation for all components, (4) backpropagation via `total_loss.backward()`, and (5) parameter update via `optimizer.step()`. Outputs include concentration profiles at $t = 200, 400, 600, 800, 1000$ days for comparison with the analytical solution.

3 Analytical Solution

The Ogata–Banks analytical solution [1] provides a theoretical reference for visual comparison:

$$C(x,t) = \frac{C_0}{2}\left[\text{erfc}\left(\frac{x - Ut}{2\sqrt{Dt}}\right) + \exp\left(\frac{Ux}{D}\right)\text{erfc}\left(\frac{x + Ut}{2\sqrt{Dt}}\right)\right] \tag{1}$$

The analytical solution is not used anywhere in the training process. It is imported from `analytical_solution.py` solely for post-training comparison and as a visual reference.

4 Results

Figure 1 presents concentration profiles along the domain at different times, comparing PINN predictions (solid lines) with the analytical Ogata–Banks solution (dashed lines). As time increases from 200 to 1000 days, the contaminant plume advects downstream while spreading due to dispersion, with the analytical solution exhibiting relatively sharp fronts followed by near-zero concentrations. While the PINN captures the overall trend of plume migration and spreading, clear differences are observed. Beyond smoother transitions in regions of steep gradients, the PINN systematically predicts an earlier reduction in concentration than the analytical solution, resulting in a left-shifted plume front and premature attenuation of contamination. In addition, the PINN slightly overestimates the inlet concentration at early distances, producing values exceeding the imposed maximum of ($C_0 = 5$), which is physically inaccurate. These behaviors indicate a systematic bias beyond simple gradient smoothing effects and suggest that the current baseline model requires further improvement; one plausible idea is to penalize nonphysical spatial increases by enforcing a constraint such as ($\partial C/\partial x \leq 0$).

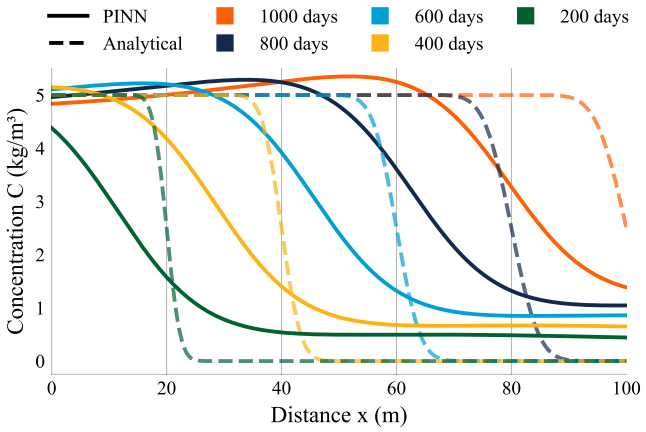

Figure 1: Concentration profiles comparing PINN predictions (solid lines) with analytical solutions (dashed lines) at $t = 200, 400, 600, 800, 1000$ days.

Figure 2 shows PINN concentration profiles at 300 and 700 days alongside the spatial distribution of the PDE residual computed directly from the learned solution, without reference to the analytical model. Although the predicted concentration profiles do not perfectly match the analytical solution, the residual plot provides valuable diagnostic information: the largest PDE violations are spatially localized, primarily around the advancing plume fronts where gradients are strongest. In contrast, upstream regions with near-constant concentration and downstream zones with near-zero concentration exhibit much smaller residuals, indicating better local satisfaction of the governing equation. This separation between solution mismatch and residual magnitude demonstrates that, even with an imperfect fit, the PINN reveals where the physics is least well enforced, which could potentially guide actionable adjustments to improve training and model performance.

5 Next Steps and Ongoing Work

Current limitations include convergence behavior and fit quality in regions with sharp gradients. Ongoing work focuses on: (1) exploring different loss weight configurations to improve convergence, (2) exploring different physical parameters (varying U, D, and L), (3) conducting parameter sensitivity studies to understand the network's response to physical parameter variations, and (4) implementing residual-based adaptive refinement [2] to improve sampling efficiency and accuracy.

References

[1] Akio Ogata and Robert B. Banks. *A Solution of the Differential Equation of Longitudinal Dispersion in Porous Media*. Professional Paper 411-A. U.S. Geological Survey, 1961. URL: https://pubs.usgs.gov/pp/0411a/report.pdf (visited on 01/01/2025).

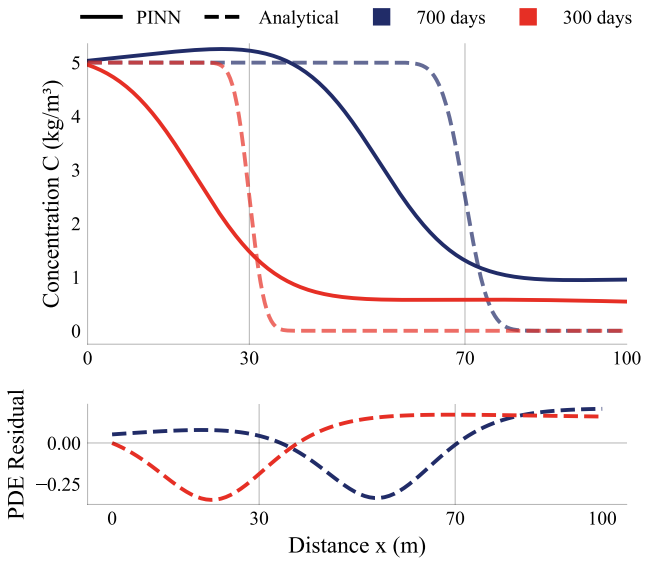

Figure 2: Concentration profiles comparing PINN predictions (solid lines) with analytical solutions (dashed lines) at $t = 300, 700$ days (top), and corresponding PDE residual distribution (bottom).

[2] Chenxi Wu et al. "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 403 (Jan. 2023), p. 115671. ISSN: 00457825. DOI: 10.1016/j.cma.2022.115671. arXiv: 2207.10289[physics]. URL: http://arxiv.org/abs/2207.10289 (visited on 12/18/2025).